2001, whose inventors are Peter Geiger, Manuel J. Alvarez II, and Thomas A. Dye, is hereby incorporated by reference in its entirety as though fully and completely set forth herein.

U.S. patent application Serial no. 10/044,786 titled "Parallel Compression And Decompression System And Method Having Multiple Parallel Compression And Decompression Engines", and filed January 11, 2002, currently herewith, whose inventors are Peter D. Geiger; Manuel J. Alvarez II; Thomas A. Dye, is hereby incorporated by reference in its entirety as though fully and completely set forth herein.

In The Claims

CLEAN COPY OF PENDING CLAIMS

Please add claims 71 and 72.

The following is a clean version of the entire set of pending claims (unamended claims appear in smaller print).

1. A data compression system comprising:

a first plurality of compression engines, wherein each of the first plurality of compression engines implements a different respective data compression algorithm, wherein at least one of the first plurality of compression engines implements a parallel data compression algorithm, and wherein each of the first plurality of compression engines is configured to:

receive uncompressed data; and

compress the uncompressed data using its respective data compression algorithm;

wherein the first plurality of compression engines produce a plurality of different versions of the compressed data;

decision logic coupled to the first plurality of compression engines and configured to:

select one of the plurality of different versions of the compressed data based upon one or more predetermined metrics; and

output the selected compressed data.

2. The data compression system of claim 1, wherein the at least one of the first plurality of compression engines implements a parallel lossless data compression algorithm.

3. The data compression system of claim 1, wherein the at least one of the first plurality of compression engines implements a parallel dictionary-based data compression algorithm.

4. The data compression system of claim 1, wherein the at least one of the first plurality of compression engines implements a parallel statistical data compression algorithm.

5. The data compression system of claim 1, wherein the at least one of the first plurality of compression engines implements a parallel data compression algorithm based on a Lempel-Ziv (LZ) algorithm.

6. The data compression system of claim 1, wherein the uncompressed data comprises a plurality of symbols, wherein the at least one of the first plurality of compression engines is operable to compare each of a plurality of received symbols with each of a plurality of entries in a history table concurrently.

7. The data compression system of claim 1,

wherein the at least one of the first plurality of compression engines comprises:

an input for receiving the uncompressed data, wherein the uncompressed data comprises a plurality of symbols, wherein the plurality of symbols includes a first symbol, a last symbol, and one or more middle symbols;

a history table comprising entries, wherein each entry comprises at least one symbol;

a plurality of comparators for comparing the plurality of symbols with entries in the history table, wherein the plurality of comparators are operable to compare each of the plurality of symbols with each entry in the history table concurrently, wherein the plurality of comparators produce compare results;

match information logic coupled to the plurality of comparators for determining match information for each of the plurality of symbols based on the compare results, wherein the match information logic is operable to determine if a contiguous match occurs for one or more of the one or more middle symbols that does not involve a match with either the first symbol or the last symbol; and

an output coupled to the match information logic for outputting compressed data in response to the match information.

8. The data compression system of claim 1, wherein each of the first plurality of compression engines implements a parallel data compression algorithm.

9. The data compression system of claim 8,

wherein each of the first plurality of compression engines implements a parallel dictionary-based data compression algorithm.

10. The data compression system of claim 9,

wherein a first compression engine implements a parallel dictionary-based data compression algorithm using tag based encoding;

wherein a second compression engine implements a parallel dictionary-based data compression algorithm using escape characters to differentiate between compressed and raw data sequences in the compressed data.

11. The data compression system of claim 8,

wherein each of the first plurality of compression engines comprises:

an input for receiving the uncompressed data, wherein the uncompressed data comprises a plurality of symbols, wherein the plurality of symbols includes a first symbol, a last symbol, and one or more middle symbols;

a history table comprising entries, wherein each entry comprises at least one symbol;

a plurality of comparators for comparing the plurality of symbols with entries in the history table, wherein the plurality of comparators are operable to compare each of the plurality of symbols with each entry in the history table concurrently, wherein the plurality of comparators produce compare results;

match information logic coupled to the plurality of comparators for determining match information for each of the plurality of symbols based on the compare results, wherein the match information logic is operable to determine if a contiguous match occurs for one or more of the one or more middle symbols that does not involve a match with either the first symbol or the last symbol; and

an output coupled to the match information logic for outputting compressed data in response to the match information.

12. The data compression system of claim 1,

wherein the one or more predetermined metrics includes compression ratio.

13. The data compression system of claim 1,

wherein the one or more predetermined metrics includes compression speed.

14. The data compression system of claim 1, wherein, in selecting one of the plurality of different versions of the compressed data, the decision logic is further configured to select a version of the compressed data with a highest compression ratio from among the plurality of versions of the compressed data.

15. The data compression system of claim 1, wherein the decision logic is further configured to:

determine a particular data compression algorithm used in the compression of the selected compressed data; and

output information indicating the particular data compression algorithm used in the compression of the selected compressed data.

16. The data compression system of claim 15, wherein the decision logic is further configured to:

encode the information indicating the particular data compression algorithm used in the compression of the selected compressed data into the selected compressed data.

17. The data compression system of claim 1, wherein a first compression engine of the first plurality of compression engines comprises a second plurality of compression engines, wherein each of the second plurality of compression engines implements a first data compression algorithm;

wherein the second plurality of compression engines are configured to compress a plurality of portions of the uncompressed data using the first data compression algorithm to produce a plurality of compressed portions of the uncompressed data; and

wherein the first compression engine of the first plurality of compression engines is configured to merge the plurality of compressed portions of the uncompressed data to produce a version of the compressed data.

18. The data compression system of claim 17, wherein the second plurality of compression engines is configured to compress the plurality of portions of the uncompressed data in a parallel fashion.

19. A data compression system comprising:

a plurality of compression engines, wherein each of the plurality of compression engines implements a different data compression algorithm, wherein at least one of the plurality of compression engines implements a parallel lossless data compression algorithm;

and wherein the plurality of compression engines are configured to:

receive uncompressed data; and

compress the uncompressed data using the different data compression algorithms to produce a plurality of different versions of the compressed data;

decision logic coupled to the plurality of compression engines and configured to:

select one of the plurality of different versions of the compressed data as a most qualified version of the compressed data based upon one or more predetermined metrics; and

output the most qualified version of the compressed data.

20. The data compression system of claim 19, wherein the at least one of the plurality of compression engines implements a parallel dictionary-based data compression algorithm.

21. The data compression system of claim 20, wherein the uncompressed data comprises a plurality of symbols, wherein the at least one of the plurality of compression engines is operable to compare each of a plurality of received symbols with each of a plurality of entries in a history table concurrently.

22. The data compression system of claim 19,

wherein the at least one of the plurality of compression engines comprises:

an input for receiving the uncompressed data, wherein the uncompressed data comprises a plurality of symbols, wherein the plurality of symbols includes a first symbol, a last symbol, and one or more middle symbols;

a history table comprising entries, wherein each entry comprises at least one symbol;

a plurality of comparators for comparing the plurality of symbols with entries in the history table, wherein the plurality of comparators are operable to compare each of the plurality of symbols with each entry in the history table concurrently, wherein the plurality of comparators produce compare results;

match information logic coupled to the plurality of comparators for determining match information for each of the plurality of symbols based on the compare results, wherein the match information logic is operable to determine if a contiguous match occurs for one or more of the one or more middle symbols that does not involve a match with either the first symbol or the last symbol; and

an output coupled to the match information logic for outputting compressed data in response to the match information.

23. A data compression system comprising:

a plurality of compression engines, wherein each of the plurality of compression engines implements a different data compression algorithm, wherein at least one of the plurality of compression engines implements a parallel lossless data compression algorithm;

first logic coupled to the plurality of compression engines and configured to:

receive uncompressed data;

send the uncompressed data to each of the plurality of compression engines;

wherein each of the plurality of compression engines is configured to:

receive the uncompressed data from the first logic;

compress the uncompressed data using a data compression algorithm implemented by the particular compression engine to produce a version of compressed data; and

output the compressed data;

second logic coupled to the plurality of compression engines and configured to:

receive from the plurality of compression engines a plurality of different versions of the compressed data each compressed with a different compression algorithm; and

select one of the plurality of different versions of the compressed data based upon one or more predetermined metrics; and

output the selected compressed data.

24. The data compression system of claim 23, wherein the at least one of the plurality of compression engines implements a parallel dictionary-based data compression algorithm.

25. The data compression system of claim 24, wherein the uncompressed data comprises a plurality of symbols, wherein the at least one of the plurality of compression engines is operable to compare each of a plurality of received symbols with each of a plurality of entries in a history table concurrently.

26. The data compression system of claim 23,

wherein the at least one of the plurality of compression engines comprises:

an input for receiving the uncompressed data, wherein the uncompressed data comprises a plurality of symbols, wherein the plurality of symbols includes a first symbol, a last symbol, and one or more middle symbols;

a history table comprising entries, wherein each entry comprises at least one symbol;

a plurality of comparators for comparing the plurality of symbols with entries in the history table, wherein the plurality of comparators are operable to compare each of the plurality of symbols with each entry in the history table concurrently, wherein the plurality of comparators produce compare results;

match information logic coupled to the plurality of comparators for determining match information for each of the plurality of symbols based on the compare results, wherein the match information logic is operable to determine if a contiguous match occurs for one or more of the one or more middle symbols that does not involve a match with either the first symbol or the last symbol; and

an output coupled to the match information logic for outputting compressed data in response to the match information.

27. The data compression system of claim 23,

wherein the one or more predetermined metrics includes compression ratio.

28. The data compression system of claim 23,

wherein the one or more predetermined metrics includes compression speed.

29. The data compression system of claim 23, wherein, in selecting the compressed data, the second logic is further configured to select a version of the compressed data with a highest compression ratio from among the plurality of different versions of the compressed data.

30. The data compression system of claim 23, wherein the second logic is further configured to:

determine a particular data compression algorithm used in the compression of the selected compressed data; and

output information indicating the particular data compression algorithm used in the compression of the selected compressed data.

31. A system comprising:

a processor;

a memory coupled to the processor and operable to store data for use by the processor;

data compression logic coupled to the memory, comprising:

a plurality of compression engines, wherein each of the plurality of compression engines implements a different respective data compression algorithm, wherein at least one of the plurality of compression engines implements a parallel data compression algorithm, and wherein each of the plurality of compression engines is configured to:

receive uncompressed data; and

compress the uncompressed data using its respective data compression algorithm;

wherein the plurality of compression engines produce a plurality of different versions of the compressed data;

decision logic coupled to the plurality of compression engines and configured to:

select one of the plurality of different versions of the compressed data based upon one or more predetermined metrics; and

output the selected compressed data.

32. The system of claim 31, wherein the at least one of the plurality of compression engines implements a parallel dictionary-based data compression algorithm.

33. The system of claim 32, wherein the uncompressed data comprises a plurality of symbols, wherein the at least one of the plurality of compression engines is operable to compare each of a plurality of received symbols with each of a plurality of entries in a history table concurrently.

34. The system of claim 31,

wherein the at least one of the plurality of compression engines comprises:

an input for receiving the uncompressed data, wherein the uncompressed data comprises a plurality of symbols, wherein the plurality of symbols includes a first symbol, a last symbol, and one or more middle symbols;

a history table comprising entries, wherein each entry comprises at least one symbol;

a plurality of comparators for comparing the plurality of symbols with entries in the history table, wherein the plurality of comparators are operable to compare each of the plurality of symbols with each entry in the history table concurrently, wherein the plurality of comparators produce compare results;

match information logic coupled to the plurality of comparators for determining match information for each of the plurality of symbols based on the compare results, wherein the match information logic is operable to determine if a contiguous match occurs for one or more of the one or more

middle symbols that does not involve a match with either the first symbol or the last symbol; and

an output coupled to the match information logic for outputting compressed data in response to the match information.

35. The system of claim 31,

wherein each of the plurality of compression engines implements a parallel dictionary-based data compression algorithm.

36. The system of claim 35,

wherein a first compression engine implements a parallel dictionary-based data compression algorithm using tag based encoding;

wherein a second compression engine implements a parallel dictionary-based data compression algorithm using escape characters to differentiate between compressed and raw data sequences in the compressed data.

37. The system of claim 31, wherein the decision logic is further configured to:

determine a particular data compression algorithm used in the compression of the selected compressed data; and

output information indicating the particular data compression algorithm used in the compression of the selected compressed data.

38. The system of claim 31, wherein the data compression logic is configured to write the selected compressed data to the memory.

39. The system of claim 31, wherein the system further comprises:

a plurality of decompression engines, wherein the plurality of decompression engines implement decompression algorithms corresponding to the different respective data compression algorithms of the plurality of compression engines;

decompression logic coupled to the plurality of decompression engines and configured to:

determine a first decompression engine of the plurality of decompression engines that implements a particular data compression algorithm used in compression of first compressed data; and

provide the first compressed data to the first decompression engine;

wherein the first decompression engine is configured to decompress the first compressed data to produce the uncompressed data.

40. The system of claim 39, wherein the data compression logic is further configured to write data compression information indicating the particular data compression algorithm used in the compression of the first compressed data to the memory;

wherein, in said determining the first decompression engine, the decompression logic is further configured to read the data compression information to determine the particular data compression algorithm.

41. The system of claim 31, further comprising:

a memory controller coupled to the processor and the memory and configured to control the memory, wherein the data compression logic is comprised in the memory controller.

42. The system of claim 31, wherein the data compression logic is comprised in the processor.

43. The system of claim 31, further comprising:

wherein the memory comprises one or more memory modules, wherein the data compression logic is comprised in one of the one or more memory modules.

44. The system of claim 31, further comprising a network interface device operable to interface the system to a network, wherein the data compression logic is comprised in the network interface device.

45. A method for compressing data in a system comprising a memory, the method comprising:

receiving uncompressed data;

providing the uncompressed data to a plurality of compression engines, wherein each of the plurality of compression engines implements a different respective data compression algorithm;

the plurality of compression engines compressing the uncompressed data using the different respective data compression algorithms, thereby producing a plurality of different versions of the compressed data, wherein said compressing comprises at least one of the plurality of compression engines compressing the uncompressed data using a parallel lossless data compression algorithm; and

selecting one of the plurality of different versions of the compressed data based upon one or more predetermined metrics.

46. The method of claim 45, further comprising:

outputting the selected compressed data.

47. The method of claim 45, wherein the at least one of the plurality of compression engines implements a parallel statistical data compression algorithm;

wherein the plurality of compression engines compressing the uncompressed data includes the at least one of the plurality of compression engines compressing the uncompressed data using the parallel statistical data compression algorithm.

48. The method of claim 45, wherein the at least one of the plurality of compression engines implements a parallel dictionary-based data compression algorithm;

wherein the plurality of compression engines compressing the uncompressed data includes the at least one of the plurality of compression engines compressing the uncompressed data using the parallel dictionary-based data compression algorithm.

49. The method of claim 48,

wherein the uncompressed data comprises a plurality of symbols;

wherein the at least one of the plurality of compression engines compressing the data using the parallel dictionary-based data compression algorithm comprises the at least one of the plurality of compression engines comparing each of a plurality of received symbols with each of a plurality of entries in a history table concurrently.

50. The method of claim 49,

wherein the at least one of the plurality of compression engines compressing the uncompressed data using the parallel dictionary-based data compression algorithm comprises:

receiving the uncompressed data, wherein the uncompressed data comprises a plurality of symbols, wherein the plurality of symbols includes a first symbol, a last symbol, and one or more middle symbols;

maintaining a history table comprising entries, wherein each entry comprises at least one symbol;

comparing the plurality of symbols with entries in the history table in a parallel fashion, wherein said comparing in a parallel fashion comprises comparing each of the plurality of symbols with each entry in the history table concurrently, wherein said comparing produces compare results;

determining match information for each of the plurality of symbols based on the compare results, wherein said determining match information includes determining if a contiguous match occurs for one or more of the one or more middle symbols that does not involve a match with either the first symbol or the last symbol; and

outputting compressed data in response to the match information.

51. The method of claim 45, wherein each of the plurality of compression engines implements a different parallel dictionary-based data compression algorithm;

wherein the plurality of compression engines compressing the uncompressed data includes the plurality of compression engines compressing the uncompressed data using the different parallel dictionary-based data compression algorithms.

52. The method of claim 51,

wherein the plurality of compression engines compressing the uncompressed data includes:

a first compression engine compressing the uncompressed data using a parallel dictionary-based data compression algorithm using tag based encoding; and

a second compression engine compressing the uncompressed data using a parallel dictionary-based data compression algorithm using escape characters to differentiate between compressed and raw data sequences.

53. The method of claim 45,
wherein the one or more predetermined metrics includes compression ratio.

54. The method of claim 45,
wherein the one or more predetermined metrics includes compression speed.

55. The method of claim 45, further comprising writing the selected compressed data to the memory.

56. The method of claim 45, further comprising:
incorporating data compression information into the selected compressed data, wherein the data compression information indicates a particular data compression algorithm used in the compression of the selected compressed data.

57. The method of claim 56, further comprising:
determining that the selected compressed data needs to be decompressed;
receiving the selected compressed data;
examining the data compression information to determine the particular data compression algorithm used in compressing the selected compressed data;
selecting a decompression engine from a plurality of decompression engines, wherein the selected decompression engine implements a decompression algorithm for decompressing data compressed using the particular data compression algorithm;
providing the selected compressed data to the selected decompression engine; and
the selected decompression engine decompressing the selected compressed data using the decompression algorithm to produce the uncompressed data.

58. A method for compressing data in a system comprising a memory, the method comprising:
receiving uncompressed data;
providing the uncompressed data to a plurality of compression engines, wherein each of the plurality of compression engines implements a different data compression algorithm;
the plurality of compression engines compressing the uncompressed data using a plurality of different data compression algorithms to produce a plurality of different versions of the compressed data, wherein said compressing includes at least one of the plurality of compression engines compressing the uncompressed data using a parallel lossless dictionary-based data compression algorithm; and
selecting one of the plurality of different versions of the compressed data as a most qualified version of the compressed data based upon one or more predetermined metrics.

59. The method of claim 58, wherein selecting the most qualified version of the compressed data further comprises selecting a version of the compressed data with a highest compression ratio from among the plurality of different versions of the compressed data.

60. The method of claim 58, wherein at least one of the first plurality of compression engines implements a parallel data compression algorithm based on a serial statistical data compression algorithm.

61. The method of claim 58, wherein at least one of the first plurality of compression engines implements a parallel data compression algorithm based on a Lempel-Ziv (LZ) algorithm.

62. The method of claim 58, wherein a first of the first plurality of compression engines comprises a second plurality of compression engines, wherein each of the second plurality of compression engines implements a first data compression algorithm, wherein said compressing further comprises:

providing a different portion of the uncompressed data to each of the second plurality of compression engines;

each of the second plurality of compression engines compressing the different portion of the uncompressed data using the first data compression algorithm to produce a compressed portion of the uncompressed data;

wherein the second plurality of compression engines compress a plurality of different portions of the uncompressed data to produce a plurality of compressed portions of the uncompressed data; and

merging the plurality of compressed portions of the uncompressed data to produce a version of the compressed data.

63. The method of claim 62, wherein the second plurality of compression engines is configured to compress the plurality of portions of the uncompressed data in a parallel fashion.

64. The data compression system of claim 62, wherein the first data compression algorithm is a parallel dictionary-based data compression algorithm.

65. A data compression system comprising:

a plurality of compression engines, wherein each of the plurality of compression engines implements a different respective parallel dictionary-based data compression algorithm, and wherein each of the plurality of compression engines is configured to:

receive uncompressed data, wherein the uncompressed data comprises a plurality of symbols; and

compress the uncompressed data using its respective data compression algorithm, wherein each of the plurality of compression engines is operable to compare each of a plurality of received symbols with each of a plurality of entries in a history table concurrently;

wherein the plurality of compression engines produce a plurality of different versions of the compressed data;

decision logic coupled to the plurality of compression engines and configured to:

select one of the plurality of different versions of the compressed data based upon one or more predetermined metrics; and

output the selected compressed data.

66. The data compression system of claim 65,

wherein a first compression engine implements a parallel dictionary-based data compression algorithm using tag based encoding;

wherein a second compression engine implements a parallel dictionary-based data compression algorithm using escape characters to differentiate between compressed and raw data sequences in the compressed data.

67. The data compression system of claim 65,

wherein the plurality of compression engines share a common history table.

68. The data compression system of claim 65,

wherein each of the plurality of compression engines includes its own history table.

69. The data compression system of claim 65,

wherein each of the first plurality of compression engines comprises:

an input for receiving the uncompressed data, wherein the uncompressed data comprises a plurality of symbols, wherein the plurality of symbols includes a first symbol, a last symbol, and one or more middle symbols;

a history table comprising entries, wherein each entry comprises at least one symbol;

a plurality of comparators for comparing the plurality of symbols with entries in the history table, wherein the plurality of comparators are operable to compare each of the plurality of symbols with each entry in the history table concurrently, wherein the plurality of comparators produce compare results;

match information logic coupled to the plurality of comparators for determining match information for each of the plurality of symbols based on the compare results, wherein the match information logic is operable to determine if a contiguous match occurs for one or more of the one or more middle symbols that does not involve a match with either the first symbol or the last symbol; and

an output coupled to the match information logic for outputting compressed data in response to the match information.

70. The data compression system of claim 65,

wherein the one or more predetermined metrics comprise at least one of compression ratio and compression speed.

71. (New) The data compression system of claim 19, wherein at least one of the plurality of compression engines performs parallel data compression using a plurality of parallel compression engines each operating in parallel on a different respective portion of said uncompressed data.

72. (New) The method of claim 58, wherein at least one of the plurality of compression engines performs parallel data compression using a plurality of parallel compression engines each operating in parallel on a different respective portion of said uncompressed data.